ONEPRED: NEXT-QUERY PREDICTION VIA RECURSIVE INTENT MEMORY IN MULTI-TURN CONVERSATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based conversational systems remain largely reactive: they respond after the user types a query. A key step toward proactive interaction is next-query prediction, which anticipates the user's subsequent query from the preceding dialogue. This task poses an efficiency–quality trade-off: concatenating the full history increases input-token consumption as the conversation grows, while using only the latest turn discards cross-turn context. Our key insight is to track the user's evolving intent trajectory across topics, unresolved needs, and interest shifts, rather than repeatedly re-read the raw history. We propose OnePred, which maintains a recursively updated textual memory as its sole cross-turn context. A two-stage reinforcement learning pipeline first teaches what to predict, then what to preserve, shaping the memory into a prediction-oriented intent chain. We introduce NQP-Bench, spanning three sources of conversations selected for context-grounded predictability, and complement it with deployment-log evaluation without predictability filtering. OnePred improves prediction quality over current-turn and full-history inputs across the evaluated model regimes, with smaller gains over matched learned-summary controls. Its input-token advantage reaches $22\times$ at long turns; accounting for generated memory yields total-token ratios of 4.0–$13.5\times$ across dialogue-length groups.

1 INTRODUCTION

LLM-based conversational systems support large-scale multi-turn interaction (Zheng et al., 2024), yet they remain largely *reactive*: the model responds only after the user types a query. We study *next-query prediction*, the task of anticipating a user's next query from the preceding dialogue. Such predictions could help users articulate follow-up questions and enable speculative execution: prefetching documents or preparing candidate answers before the next query arrives. These applications motivate learning to recognize the user's evolving intent across turns.

Next-query prediction differs from related problems in conversational systems. *Query suggestion* often optimizes clicks or preferences (Min et al., 2025; Yin et al., 2025), while *proactive dialogue* focuses on system actions that steer a conversation toward a goal (Deng et al., 2023; Wu et al., 2019). Here, the target is the user's next query. Evaluation is challenging because a conversation may support several plausible continuations, and many real-world queries cannot be inferred from prior context alone.

Two straightforward approaches to handling dialogue history each fall short. A *Current-turn* model observes only the latest exchange. While efficient, it can miss topic continuations or recurring needs that span earlier turns. In contrast, a *Full-history* model concatenates all previous turns, providing rich context at an increasing input-token cost. Predictive signals may also be buried under lengthy assistant responses and topical digressions. The core challenge is therefore one of compression: effective prediction requires cross-turn context, but only the subset that is informative for the user's next query.

Our key insight is to represent this context as an *intent chain*: the evolving trajectory of topics, unresolved needs, and interest shifts. We use a bounded free-form text state, allowing the LLM to read and write memory natively without additional modules while keeping its content interpretable. We

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

propose **OnePred**, in which this memory is recursively updated at each turn. The model receives only the previous memory and the current user–assistant exchange, and decides what to retain, update, or discard. The memory carries cross-turn context while keeping the historical component of the input bounded.

Learning to maintain such memory is non-trivial. Without annotated memory states, we train memory updates through their contribution to next-query prediction. This couples two capabilities: recognizing useful predictive signals and retaining them under a memory budget. We adopt a two-stage reinforcement learning pipeline. **Stage 1** (Full-History RL) gives the model the complete conversation and trains next-query prediction directly. **Stage 2** (Agentic Memory RL) removes raw-history access and trains the model to predict through its memory. This curriculum first teaches what to predict, then what to preserve.

To evaluate these capabilities, we construct **NQP-Bench** from three conversation sources. It selects examples whose recorded next query is judged predictable from context; a separate deployment-log test omits this predictability filter. Our contributions are threefold:

- **Benchmark.** NQP-Bench provides a testbed for context-grounded next-query prediction across private, public, and cross-source settings, with graded intent evaluation and human assessment.
- **Method.** OnePred combines recursive intent memory with a two-stage RL curriculum for learning prediction and memory updates without annotated memory states.
- **Empirical findings.** OnePred improves alignment over basic history interfaces and achieves smaller gains over learned-summary controls. Token accounting includes generated memory, and deployment-log evaluation tests the method beyond predictability-selected examples.

2 NQP-Bench and Evaluation Scope

Given a prefix $\mathcal{C}_T = ((q_1, r_1), \ldots, (q_T, r_T))$, the task is to generate candidates for the recorded next user query q_{T+1}. This reference is one observed continuation, not the unique valid continuation. We use three sources: NQP-Priv from a commercial deployment, NQP-Wild from WildChat (Zhao et al., 2024), and NQP-Share from ShareChat (Yan et al., 2025). NQP-Share is reserved for cross-source evaluation. NQP-Priv is not released; its implications for reproduction are discussed in the Reproducibility Statement.

Table 1: NQP-Bench sizes. All subsets are filtered for context-grounded predictability.

Subset	Train	Test	Avg. turns	Source
NQP-Priv	21,707	2,212	4.79	Private logs
NQP-Wild	15,916	2,354	4.82	WildChat
NQP-Share	–	1,947	5.57	ShareChat

Construction and selection. Figure 1 summarizes the pipeline. Heuristic filtering restricts language, conversation and query lengths, removes degenerate examples, and deduplicates the public data. GLM-5 (GLM-5-Team et al., 2026) then evaluates safety, predictability, clarity, and repetition, returning KEEP, UNCERTAIN, or DROP. Gemini 2.5 Pro (Comanici et al., 2025) reviews KEEP and UNCERTAIN decisions. For rejected conversations with at least four turns, truncation mining searches backward for a predictable continuation and retains the corresponding prefix. A final pass removes near-duplicates and over-length samples before splitting.

Selection is substantial: Stage II forwards 29.2K of 148K heuristically retained WildChat examples and 6.1K of 30.5K ShareChat examples, approximately 20% in each case. The other approximately 80% fail the *combined* criteria for safety, predictability, clarity, and repetition. Stage III retains 21.0K and 4.7K, respectively, before mining and final filtering (Appendix C). NQP-Bench therefore measures prediction on selected, context-grounded continuations. Its cross-source evaluation applies the same selection policy to a held-out source; the separate deployment-log evaluation extends testing beyond this policy.

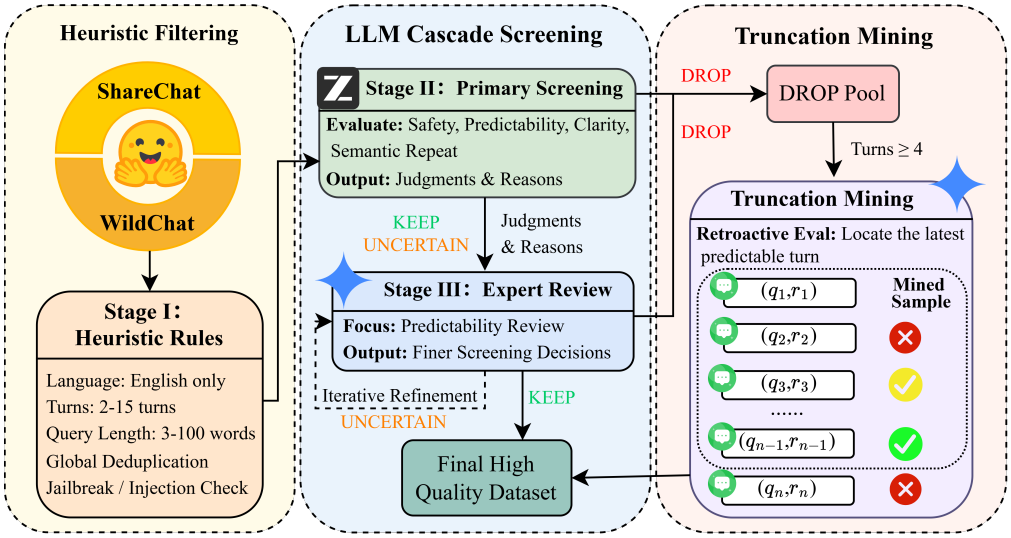


Figure 1: NQP-Bench construction pipeline. Offline selection evaluates the predictability of the recorded next query from its dialogue prefix.

Judges rate candidate–reference intent alignment on a five-level scale, normalized as $S(\hat{q}, q) = 25(s-1)$ for $s \in \{1, \ldots, 5\}$. Appendix H gives the rubric. Scores measure agreement with one observed continuation; a plausible alternative can receive a low score.

3 OnePred: Prediction Through Bounded Memory

3.1 Recursive Intent Memory

For observation $o_t = (q_t, r_t)$ and initially empty m_0, the model generates a textual memory and N candidate queries:

$$(\widetilde{m}_t, \hat{q}_{t+1}^{(1)}, \ldots, \hat{q}_{t+1}^{(N)}) = f_\theta(m_{t-1}, o_t), \qquad m_t = \mathrm{Trunc}_k(\widetilde{m}_t). \tag{1}$$

The subscript $t+1$ denotes the future user turn. Only the first k memory tokens are passed forward; previous raw observations are not accessible. We use $k = 500$ and $N = 3$ in the reported configuration. The memory has no prescribed schema or additional learned module. We use "intent chain" to refer to the topics and interests recorded in this free-form state.

At inference the model emits memory and predictions after each exchange. During training, intermediate turns emit memory only, and the final turn also emits candidates. With fixed prompt overhead P, input length is at most $P + k + |o_t|$. The historical-state contribution is therefore bounded independently of t, while the current observation contributes $|o_t|$ tokens. The budget k applies to stored memory; generation before truncation is accounted for separately in the output-token measurements.

3.2 Two-Stage Training

Stage 1: full-history prediction. We train Qwen3 (Yang et al., 2025) to generate candidate next queries from $\mathcal{C}_T$ for 500 GRPO steps (Shao et al., 2024). This establishes predictive behavior before imposing the memory constraint. Outcome-based optimization allows training without annotated memory states.

Stage 2: recursive-memory prediction. Initialized from Stage 1, the model applies the memory update in Eq. 1 sequentially for a further 200 training steps, emitting candidates only at the final exchange. Intermediate memory updates have no direct target; the final prediction reward trains the trajectory. The objective and candidate count remain the same, while access to raw history is removed. Both stages use eight rollouts per prompt. Appendix A gives the optimizer and generation settings.

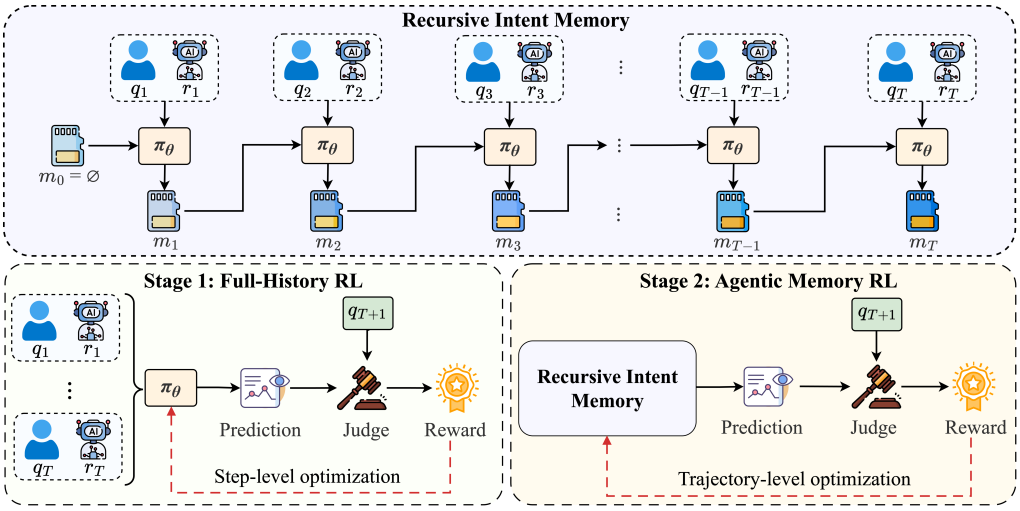


Figure 2: OnePred overview. Top: the previous memory and current exchange determine the next memory. Bottom: full-history prediction training is followed by recursive-memory training with terminal prediction feedback.

Reward and credit assignment. The prediction component takes the maximum candidate rating. With a structural-format reward, the training reward is

$$R = \lambda R_{\text{judge}} + (1-\lambda) R_{\text{format}}, \qquad \lambda = 0.9. \tag{2}$$

R_{format} penalizes malformed outputs that break the memory interface. For a group of G rollouts, the trajectory advantage is obtained by normalizing their terminal rewards:

$$\widehat{A}_i = \frac{R_i - \text{mean}_j(R_j)}{\text{std}_j(R_j)}. \tag{3}$$

For nonzero-variance groups, this normalization expresses each trajectory's reward relative to the other rollouts. For a generated action a with conditioning context h in rollout i, a clipped surrogate term is

$$\ell_i(a,h) = -\min\{\rho_\theta(a \mid h)\widehat{A}_i, \text{clip}(\rho_\theta(a \mid h), 1-\epsilon, 1+\epsilon)\widehat{A}_i\}, \tag{4}$$

where $\rho_\theta(a \mid h) = \pi_\theta(a \mid h)/\pi_{\text{old}}(a \mid h)$ and $\epsilon = 0.2$. The ratio is computed for each generated action under its conditioning context. For Stage 2, the same terminal advantage is assigned to the preceding memory actions,

$$\widehat{A}_{i,t} = \widehat{A}_i \quad (t = 1, \ldots, T). \tag{5}$$

Each memory update thus receives feedback from the final prediction outcome. This shared signal trains the sequence without intermediate memory annotations; action-specific credit remains unresolved.

4 Experiments

4.1 Setup and Evaluation Protocol

We compare Current-turn, Full-history, and recursive-memory interfaces with Gemini-3.1-Pro (Google DeepMind, 2026), untrained Qwen3-8B, and task-trained Qwen3-8B. The Gemini and base-model memory rows evaluate the recursive interface through prompting alone. Current-turn and Full-history RL each use Stage 1 with their respective inputs; OnePred uses both stages. Training uses the NQP-Priv and NQP-Wild training splits. NQP-Share is never used for training.

All interfaces generate three candidates with greedy decoding. Let $S_i = S(\hat{q}^{(i)}, q_{T+1})$. We report $S_{\text{Top1}} = S_1$, $S_{\text{Mean3}} = \frac{1}{3}\sum_i S_i$, and $S_{\text{Best3}} = \max_i S_i$. Best-of-3 measures candidate-set coverage by selecting the highest alignment score against the observed future reference. Reported evaluation scores use intent alignment alone; the format reward is confined to training.

The primary ensemble comprises Claude Opus 4.7 (Anthropic, 2026), GPT-5.5 (OpenAI, 2026), and Gemini-3.1-Pro. Each rates candidate–reference pairs; the majority rating is used, or the median if all three disagree. The same ensemble provides RL rewards, so primary evaluation is not independent of optimization. Five human annotators independently score 250 sampled instances per subset. Human evaluation uses the same five-level intent rubric. Human agreement (Fleiss' κ) is 0.72/0.75/0.70 and human–judge Spearman correlation is 0.78/0.81/0.76 on Priv/Wild/Share. Appendix H details these agreement measures and their scope.

4.2 Prediction Quality and Learned-Compression Controls

Table 2: Best-of-3 results on the selected NQP-Bench subsets. Scores are normalized intent ratings, not prediction accuracy. For each subset, we report both the LLM-judge score (Judge) and the human evaluation score (Human). Human scores use 250 sampled instances per subset; Judge scores use the test sets. The last two columns report unweighted means across the three subsets, rounded to two decimal places.

Method	NQP-Priv		NQP-Wild		NQP-Share		Average	
	Judge	Human	Judge	Human	Judge	Human	Judge	Human
Gemini-3.1-Pro								
Current-turn	41.18	42.10	45.53	45.30	42.78	43.30	43.16	43.57
Full-history	40.81	41.20	48.81	49.90	46.30	46.10	45.31	45.73
Ours	**44.22**	**44.90**	**50.72**	**51.10**	**48.77**	**49.70**	**47.90**	**48.57**
Base Qwen								
Current-turn	28.92	30.10	36.09	36.40	35.02	35.90	33.34	34.13
Full-history	31.37	31.90	38.20	39.20	35.70	36.30	35.09	35.80
Ours	**33.47**	**33.20**	**39.57**	**40.40**	**37.84**	**37.60**	**36.96**	**37.07**
RL-trained Qwen								
Current-turn	38.17	38.60	44.02	44.20	40.50	40.20	40.90	41.00
Full-history	40.79	40.60	44.35	45.00	42.15	42.50	42.43	42.70
Ours	**42.15**	**42.70**	**46.00**	**47.30**	**43.84**	**45.00**	**44.00**	**45.00**

Recursive memory exceeds Full-history in all nine reported model–dataset comparisons (Table 2), with average judge margins of 2.60, 1.87, and 1.57 for Gemini, base Qwen, and RL-trained Qwen. Human scores preserve this ranking. The gains hold on both public subsets as well as NQP-Priv. Absolute alignment remains moderate under the five-level rubric.

Controls for learned compression. Two additional Qwen3-8B controls use the same 500-token memory budget, three candidates, final next-query reward, and 500+200 training steps as OnePred. Full-history Summary RL first reads the entire history, generates a bounded summary, then predicts from that summary. Summary-memory RL uses the same recursive interface but asks for a retrospective dialogue summary. Both controls receive the same next-query prediction reward. Summary-memory RL therefore tests the choice of memory instruction while keeping the training objective fixed.

OnePred's margin is 0.66 over full-history learned summarization and 0.87 over recursive learned summarization (Table 3). These gains are smaller than those over basic history interfaces. Summary-memory RL shares the recursive input structure, whereas Full-history Summary RL reads the complete history before each compression. The comparisons therefore distinguish memory instructions within a common recursive interface and history access across the two summary designs.

Candidate aggregation and evaluator agreement. Table 4 separates first-candidate alignment from candidate-set coverage.

OnePred's margin over Full-history remains positive at 1.25 for Top-1, versus 1.65 for Best-of-3. Top-1 and Best-of-3 score the same three-candidate output, using its first candidate and highest reference-alignment score, respectively.

Table 3: NQP-Wild Best-of-3 scores. Gemini and Qwen rows are separate comparisons. All learned-summary controls use the same next-query reward and candidate count as OnePred.

Model / method	Historical input	RL steps	Judge
Gemini: sliding window	Last 3 turns	–	46.14
Gemini: summarize then predict	Full history to summary	–	47.42
Gemini: recursive memory	Previous memory	–	50.72
Qwen: Full-history RL	Full history	500	44.35
Qwen: Full-history Summary RL	Full history to summary	700	45.34
Qwen: Summary-memory RL	Previous summary	700	45.13
Qwen: OnePred	Previous intent memory	700	46.00

Table 4: Candidate-count control on NQP-Wild with RL-trained Qwen3-8B. Every interface generates three candidates; Top-1 scores the first candidate from that output.

Method	Top-1	Mean-of-3	Best-of-3
Current-turn	41.83	42.76	44.02
Full-history	42.68	43.35	44.35
OnePred	**43.93**	**44.71**	**46.00**
Gain over Full-history	+1.25	+1.36	+1.65

Additional judges held out from reward computation preserve the ranking (Appendix K). Their model families overlap with the backbone and curation models. Appendix G reports marginal intervals for the dataset means and defines their statistical scope.

4.3 Training Curriculum

Table 5: NQP-Wild ablation. Every row is evaluated with recursive-memory prompting.

Training	Total steps	Judge	Human
Base Qwen	0	39.57	40.40
Stage 1 only	500	43.82	44.60
Stage 1 only, step-matched	700	43.89	44.90
Stage 2 only	200	42.96	43.40
Stage 1 then Stage 2	700	46.00	47.30

After 500 full-history steps, switching to memory training for 200 steps yields 2.11 more Judge points than continuing full-history training (Table 5). The two 700-step runs match optimization steps; token and compute budgets can differ with the history interface. Stage 2-only training also improves over the base model after 200 steps. Its shorter schedule makes it a separate training configuration rather than an initialization-only control.

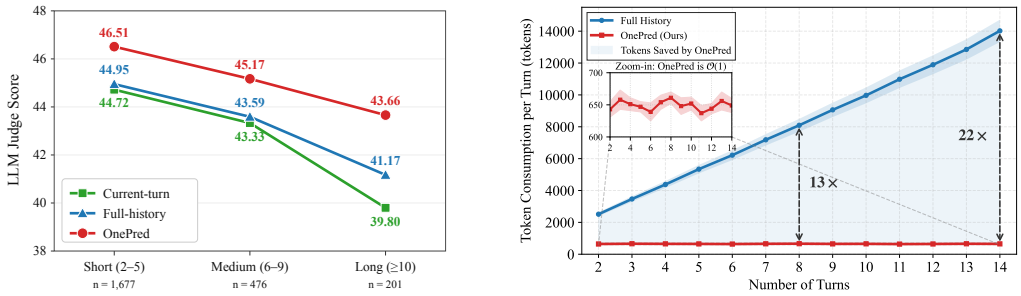
4.4 Evaluation Without Predictability Filtering

We additionally evaluate the trained Qwen3-8B models on a separate test set sampled from deployment conversations without predictability filtering: 2,000 prediction points, with a shared 250-example subset for human evaluation. All methods generate three candidates and use the same intent-rating rubric (Appendix M). Table 6 reports Top-1 alignment and Best-of-3 coverage under the metrics defined in Section 4.1.

OnePred gains 0.95 Best-of-3 Judge points over Full-history and 0.40 over Summary-memory RL; human margins are 0.40 and 0.50. OnePred has the highest reported mean under both evaluations, with smaller margins than on the selected benchmark. Model-size results appear in Appendix L.

Table 6: Offline reference-alignment scores on deployment logs sampled without predictability filtering. Human scores use Best-of-3.

	Judge		Human
Method	Top-1	Best-of-3	Best-of-3
Current-turn	29.40	31.20	31.80
Full-history	31.10	33.10	33.60
Summary-memory RL	31.55	33.65	33.50
OnePred	**32.00**	**34.05**	**34.00**



(a) Scores across three length groups.

(b) Input tokens per prediction turn.

Figure 3: Prediction quality and input-context use on NQP-Wild. (a) OnePred has the highest mean in each of the three reported length groups; these are observational strata. (b) Historical memory is capped at 500 tokens. The 22× ratio concerns input tokens, not total cost or latency.

4.5 Prediction Quality Across Dialogue Lengths

Figure 3a reports short (2–5-turn; $n = 1{,}677$), medium (6–9-turn; $n = 476$), and long (≥ 10-turn; $n = 201$) dialogues. OnePred scores 46.51/45.17/43.66, compared with 44.95/43.59/41.17 for Full-history and 44.72/43.33/39.80 for Current-turn. OnePred leads in all three reported length groups. Scores decline from short to long dialogues for each method, with a smaller endpoint decrease for OnePred. These are observational length groups whose topic and difficulty composition may vary. The complementary breakdown by difficulty and intent transfer appears in Appendix J.

4.6 Efficiency: Context Length Versus Serving Cost

When prediction runs after every turn, both interfaces make one combined prediction call per turn, but OnePred additionally generates memory. The input-only comparison in Figure 3b reaches approximately 22× at turn 14. Table 7 includes generated outputs from additional profiling; these include memory updates for OnePred and candidates for both systems.

Table 7: Per-turn token counts by dialogue length. Ratios divide Full-history total tokens by OnePred total tokens, with equal weight for input and output tokens.

Turns	Method	Input	Output	Total	Ratio
2–5	Full-history	3,612	92	3,704	3.99×
	OnePred	642	286	928	–
6–9	Full-history	7,481	94	7,575	8.03×
	OnePred	654	289	943	–
≥ 10	Full-history	12,860	96	12,956	13.54×
	OnePred	665	292	957	–

Runtime additionally depends on prefix caching and serving configuration. Appendix K reports preliminary latency measurements with and without caching. In deployment, the value of speculative prediction also depends on how often its outputs are used.

4.7 What the Memory Retains

Figure 4 (discussed further in Appendix D) illustrates the distinction between retaining conversational facts and retaining a predictive trajectory. Across 13 turns, the user moves from nutrition to health questions, recipes, and skin collagen. The memory links the recurring interest in food and nutrition to the recent collagen topic. Its first candidate, "What foods are best for increasing collagen production?", matches the recorded next-query intent. A current-turn view lacks the earlier food thread, while full history retains that thread together with intervening detail. This example illustrates how memory can connect an earlier interest to the current topic. The case-study compression ratio is 52-fold in characters; Section 4.6 separately reports token consumption across dialogue-length groups.

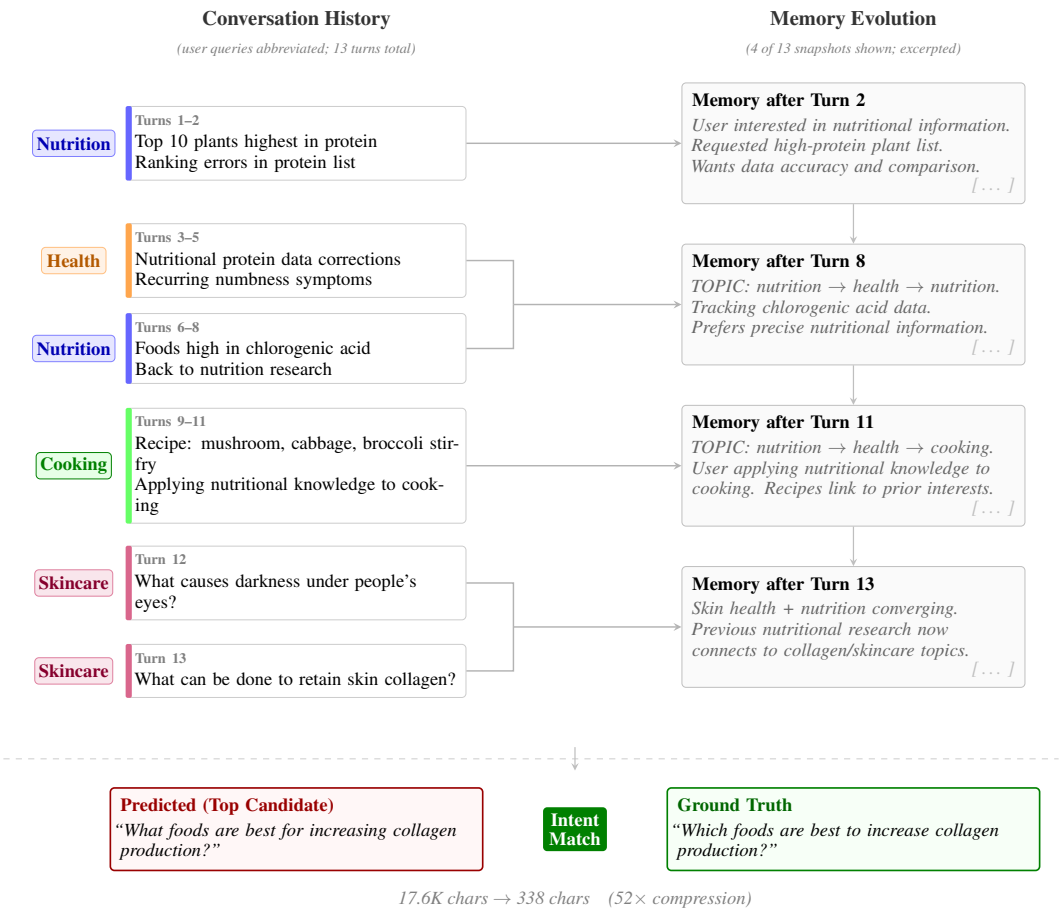


Figure 4: Qualitative example from NQP-Wild. Left: abbreviated user queries from a 13-turn conversation spanning four topic clusters (assistant responses omitted). Right: four representative snapshots of the recursive memory (excerpted; the full memory is updated at every turn). The memory compresses 17.6K characters into 338 characters (52×) while preserving the cross-topic intent trajectory, with its top candidate matching the ground-truth next-query intent.

5 Related Work

Query suggestion and proactive dialogue. Query suggestion (QS) has evolved from traditional search environments (Baeza-Yates et al., 2004; Cao et al., 2008; Sordoni et al., 2015; Dehghani et al., 2017) to conversational AI. Recent generative systems address domain-specific query suggestion in e-commerce (Guo et al., 2026) and local services (Chen et al., 2026), as well as conversational search guided by click and preference signals (Min et al., 2025; Yin et al., 2025). These settings include completing a typed query prefix, which differs from anticipating an unobserved future query. OnePred evaluates alignment with a recorded next query on a predictability-selected benchmark and a separate deployment-log sample.

Proactive dialogue systems typically steer conversations toward system-defined goals, such as task completion (Deng et al., 2023; Wu et al., 2019). This line of work primarily studies the next system action, whereas next-user-utterance prediction is a related but distinct objective. ProUtt (Wang et al., 2025) is particularly close to our task: it constructs preference data from offline-synthesized intent trees to predict the next user utterance. OnePred studies the complementary problem of retaining predictive context in a bounded recurrent state, coupling next-query prediction with sequential memory updates.

Learned memory and reinforcement learning. Access to a long context does not ensure effective use of its contents. Liu et al. (2024) find that retrieval and question-answering performance depends on where relevant information appears, with degradation when it lies in the middle of the context. These findings motivate selective context construction when relevant information is dispersed across a long interaction. LongMem (Wang et al., 2023) takes a retrieval approach: a frozen backbone encodes past context, while an adaptive side-network retrieves and reads cached representations. OnePred instead carries a generated textual state between turns, without a separate retrieval network or access to earlier raw exchanges. MemAgent reads segmented text and repeatedly overwrites its memory through RL; MEM1 learns a compact state that combines memory and reasoning across multi-turn tasks (Yu et al., 2026; Zhou et al., 2025). OnePred shares outcome-trained selective retention with these methods, but the prediction target is an unobserved future user query. Its prediction-oriented instruction asks the model to preserve evolving interests and unresolved needs as clues to likely continuations.

GRPO-style objectives (Shao et al., 2024; DeepSeek-AI et al., 2025) optimize generated trajectories using outcome feedback. OnePred applies a shared terminal advantage to sequential memory updates without intermediate supervision, following a curriculum from full-history prediction to recursive-memory prediction.

6 Limitations

NQP-Bench focuses on continuations judged predictable from context. The deployment-log evaluation broadens this scope, but results remain specific to the sampled conversation sources. A single recorded continuation also leaves other plausible next queries uncredited. Bounded memory can discard details that later become relevant, consistent with the weaker Associated Shift results. Comparisons cover basic history interfaces and learned-summary controls; direct evaluations against ProUtt, MemAgent, and MEM1 remain future work. The reported gains are point estimates without repeated-seed evaluation or paired significance tests. These limitations motivate broader evaluation and uncertainty-aware memory designs.

7 Conclusion

OnePred uses bounded textual memory to predict the next user query without revisiting the full conversation. On NQP-Bench, it improves intent alignment over current-turn and full-history inputs, with smaller gains over matched learned-summary controls. OnePred also achieves the highest reported means in the deployment-log evaluation without predictability filtering. Including memory generation in token accounting still yields substantial reductions in predictor token use. Future work should test when to abstain, compare additional memory methods, and measure the benefit of predictions in downstream applications.

ETHICS STATEMENT

We use WildChat and ShareChat under their respective terms. Private logs were de-identified, reviewed under internal data-governance policy, and are not released; only aggregate results are reported. Human annotators were informed of the evaluation task and compensated at local market rates. Next-query prediction can expose sensitive interests or enable unwanted profiling; deployment should provide consent, opt-out, data minimization, and safeguards against manipulation. Benchmark predictability selection should not be interpreted as permission to infer unexpressed personal attributes.

REPRODUCIBILITY STATEMENT

Appendix A summarizes the training and inference settings. Sections 2 and 4.1 describe benchmark construction and evaluation, while Appendix F reports sensitivity to memory budget, candidate count, and reward weight. The anonymous code repository is `https://anonymous.4open.science/r/OnePred-21F4`. NQP-Priv is not publicly released. Consequently, experiments involving its training split cannot be reconstructed solely from the public benchmark subsets; this limits independent reproduction of mixed-data training.

AI USE STATEMENT

The study uses language models for predictability screening, dataset annotation, training rewards, and evaluation as described in the paper. During manuscript revision, an AI assistant supported literature discovery, language editing, methodological discussion, interpretation of results, numerical analysis and consistency checks, and LaTeX formatting. The assistant did not conduct model-training or prediction experiments. The authors checked numerical revisions against their experimental records, reviewed the AI-assisted work, and take responsibility for the final manuscript and its empirical claims.

REFERENCES

Anthropic. Claude Opus 4.7 System Card. Anthropic system card, apr 2026. Model system card for Claude Opus 4.7.

Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In *Current Trends in Database Technology – EDBT 2004 Workshops*, pp. 588–596. Springer, 2004. doi: 10.1007/978-3-540-30192-9_58.

Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 875–883, 2008. doi: 10.1145/1401890.1401995.

Jinwen Chen, Shiwen Zhang, Shuai Gong, Zheng Zhang, Yachao Zhao, Lingxiang Wang, Haibo Zhou, Wei Lin, and Hainan Zhang. LocalSUG: City-Preference-Enhanced LLM for Query Suggestion in Local-Life Services. *arXiv preprint arXiv:2603.04946v3*, 2026. URL `https://arxiv.org/abs/2603.04946v3`.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. URL `https://arxiv.org/abs/2507.06261`.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025. URL `https://arxiv.org/abs/2501.12948`.

Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2017. URL https://arxiv.org/abs/1708.03418.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*, 2023. URL https://arxiv.org/abs/2305.02750.

GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, et al. GLM-5: from Vibe Coding to Agentic Engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

Google DeepMind. Gemini 3.1 pro model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf, 2026. Accessed: 2026-09-25.

Xian Guo, Ben Chen, Siyuan Wang, Ying Yang, Mingyue Cheng, Chenyi Lei, Yuqing Ding, and Han Li. OneSug: The Unified End-to-End Generative Framework for E-commerce Query Suggestion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(17):14774–14782, 2026. doi: 10.1609/aaai.v40i17.38497.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638.

Erxue Min, Hsiu-Yuan Huang, Xihong Yang, Min Yang, Xin Jia, Yunfang Wu, Hengyi Cai, Junfeng Wang, Shuaiqiang Wang, and Dawei Yin. CTR-Guided Generative Query Suggestion in Conversational Search. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2624–2634, 2025. doi: 10.18653/v1/2025.emnlp-industry.178.

OpenAI. GPT-5.5 System Card. https://openai.com/index/gpt-5-5-system-card/, 2026. Accessed: 2026-09-25.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv:2409.19256*, 2024. URL https://arxiv.org/abs/2409.19256.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 553–562, 2015. URL https://arxiv.org/abs/1507.02221.

Jinqiang Wang, Huansheng Ning, Jianguo Ding, Tao Zhu, Liming Chen, and Chris Nugent. LLM-Driven Preference Data Synthesis for Proactive Prediction of the Next User Utterance in Human-Machine Dialogue. *arXiv preprint arXiv:2601.09713*, 2025. URL https://arxiv.org/abs/2601.09713.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, 2023.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3794–3804, 2019. doi: 10.18653/v1/P19-1369.

Yueru Yan, Tuc Nguyen, Bo Su, Melissa Lieffers, and Thai Le. ShareChat: A Dataset of Chatbot Conversations in the Wild. *arXiv preprint arXiv:2512.17843*, 2025. URL https://arxiv.org/abs/2512.17843.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Junhao Yin, Haolin Wang, Peng Bao, Ju Xu, and Yongliang Wang. From Clicks to Preference: A Multi-stage Alignment Framework for Generative Query Suggestion in Conversational System. *arXiv preprint arXiv:2508.15811*, 2025. URL https://arxiv.org/abs/2508.15811.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-Based Memory Agent. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.02259.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=BOfDKxfwt0.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. *arXiv preprint arXiv:2506.15841*, 2025. URL https://arxiv.org/abs/2506.15841.

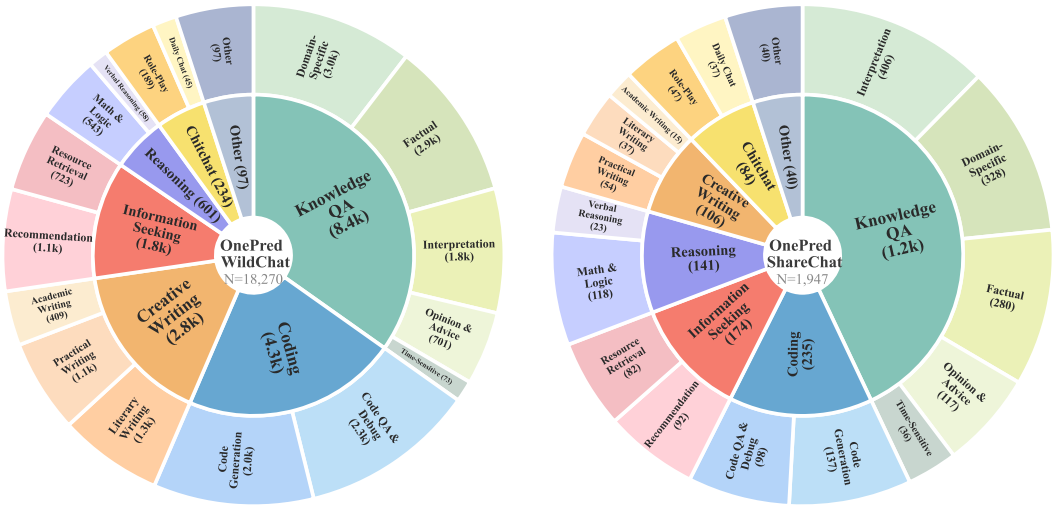
A Implementation Details

Training setup. The backbone model is Qwen3-8B, trained with the verl framework (Sheng et al., 2024) for GRPO. Both stages use the AdamW optimizer with a PPO clip ratio $\epsilon = 0.2$ (Eq. 4). **Stage 1** (Full-History RL): 500 steps, constant learning rate 1×10^{-6}, batch size 32, 8 rollouts per prompt, maximum prompt length 8,192, maximum response length 3,072. **Stage 2** (Agentic Memory RL): initialized from the Stage 1 checkpoint, 200 steps, learning rate 5×10^{-7} with cosine schedule and 2% warmup, batch size 32, 8 rollouts per prompt, maximum response length 4,096. The increased response length in Stage 2 accommodates the multi-turn trajectory format (memory update at each intermediate turn, with prediction appended only at the final turn). The reward follows Eq. 2 with $\lambda = 0.9$; the memory budget is $k = 500$ tokens per turn; the number of candidate predictions is $N = 3$. Sensitivity analyses for k, N, and λ are provided in Appendix F. For the scaling experiments (Table 20), the 1.7B and 4B models are trained with the same hyperparameters as the 8B model.

Compute budget. Stage 1 and Stage 2 training were conducted on 8 NVIDIA A100 80GB GPUs, taking approximately 256 GPU-hours in total for the 8B model (including both stages and all rollout generation).

Inference. During evaluation, all models decode greedily (temperature 0). The memory budget is enforced by hard truncation at k tokens after each memory update; if the model's output exceeds k tokens in the memory field, the suffix beyond k tokens is discarded before the next turn.

Evaluation details. The LLM Judge score is determined by majority vote of three frontier models: Claude Opus 4.7 (Anthropic, 2026), GPT-5.5 (OpenAI, 2026), and Gemini-3.1-Pro (Google DeepMind, 2026). Each judge independently rates every candidate–ground-truth pair on the 5-point rubric (Table 14), and the majority rating is taken as the final score (median when no majority exists), linearly mapped to $[0, 100]$ via $(s-1)/4 \times 100$ where $s \in \{1, 2, 3, 4, 5\}$. The same ensemble is used for the training reward R_{judge}. For human evaluation, five expert annotators independently scored 250 randomly sampled instances per subset on the same rubric, and their majority vote is taken as the final human score. Further validation of the judge protocol is provided in Appendix H.



(a) Intent Distribution (NQP-Wild) (b) Intent Distribution (NQP-Share)

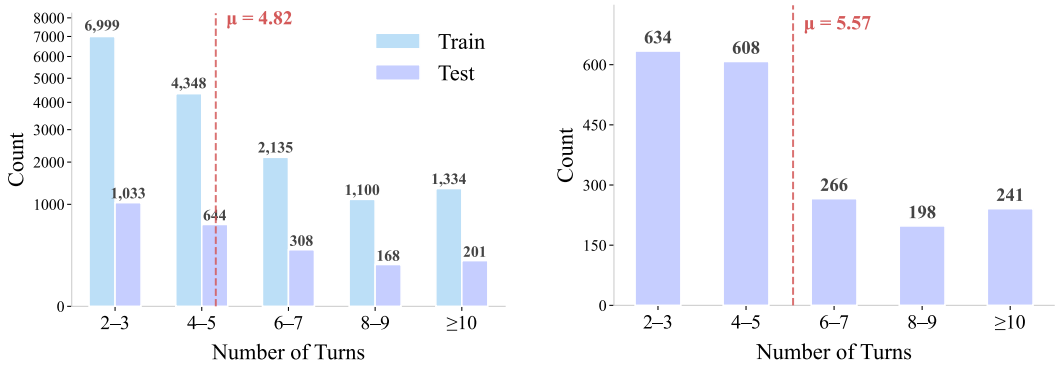
Figure 5: Hierarchical intent categories in the selected public subsets. Numeric labels give category counts, with large counts abbreviated in thousands; sector sizes provide a visual summary.

B Dataset Statistics and Characteristics

This section presents the statistical characteristics of NQP-Bench. All structured metadata used in this analysis were automatically annotated using Gemini-3.1-Pro. The intent-transfer and difficulty distributions use all curated examples: the combined training and test splits for NQP-Wild, and the held-out NQP-Share set. Model performance is evaluated on test examples only.

B.1 Intent Diversity

NQP-Bench groups intents into seven primary categories, with secondary categories and an Other category shown in the outer ring. Figure 5 illustrates the intent distribution across the WildChat and ShareChat subsets. The data span a wide spectrum of user needs, anchored by three major domains: *Knowledge QA* (focusing on factual queries and domain-specific interpretation), *Coding* (spanning code generation and debugging), and *Creative Writing*. The category distribution describes the range of intents retained by the benchmark curation procedure.



(a) Turn Distribution (NQP-Wild)

(b) Turn Distribution (NQP-Share)

Figure 6: Distribution of conversation lengths. The long-dialogue test groups contain 201/2,354 NQP-Wild examples (8.5%) and 241/1,947 NQP-Share examples (12.4%).

B.2 Conversation Depth

Figure 6 shows both short and longer conversations. The reported source-level average lengths are 4.82 (NQP-Wild) and 5.57 (NQP-Share). In the test sets, 201 Wild examples and 241 Share examples have at least ten turns. These groups permit a length-stratified comparison, although observational strata can differ in content and difficulty.

B.3 Dynamics of Intent Transfer

Users in multi-turn dialogues rarely follow a static path. We categorize the dynamic evolution of user intent into four paradigms, defined by their core behavioral characteristics:

- **Deepening:** Focuses on conceptual comprehension, where the user stays on the same core topic but goes deeper to seek a more sophisticated understanding.
- **Application:** Operates at the practical execution level, where the user performs further operations on the AI's previous output to produce a derived artifact.
- **Associated Shift:** Represents logically inferable topic transitions, where the user switches to a different but contextually connected topic.
- **Challenge:** Centers on contradiction and conflict resolution, where the user pushes back on the AI's output to correct substantive errors or flawed assumptions.

Figure 7 reveals that while *Deepening* accounts for over half of the transitions within the selected samples, approximately 40% of the queries involve *Application* or *Associated Shift*. This motivates evaluating transitions beyond direct topic continuation within the selected benchmark.

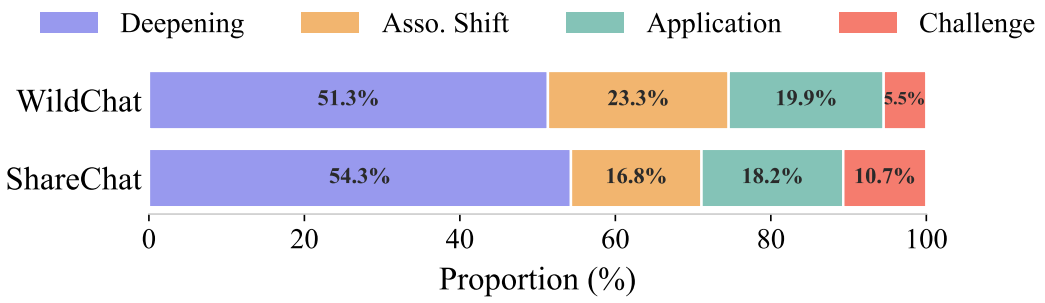


Figure 7: Distribution of intent-transfer types across the curated NQP-Wild and NQP-Share datasets.

B.4 Difficulty

We annotate prediction difficulty using three criteria: contextual distance, ambiguity among plausible continuations, and reasoning requirements. A prediction is classified as **Hard** if it triggers any of the following conditions:

- **Contextual Distance:** The prediction requires information from ≥ 2 turns ago (i.e., it cannot be predicted using only the latest exchange); for *Associated Shift*, the polarity is reversed because a distant anchor makes the shift more predictable, whereas a jump without prior grounding is inherently harder.
- **Predictive Entropy:** The current dialogue state presents ≥ 5 independent and equiprobable future directions.
- **Reasoning Gap:** Establishing the connection between the dialogue history and the target query requires implicit reasoning or external domain knowledge.

As depicted in Figure 8, under this annotation rule, 41.6% of NQP-Wild and 52.0% of NQP-Share samples are categorized as *Hard*. The Easy/Hard labels are operational categories defined by these annotation criteria and are used to stratify the evaluation in Appendix J.

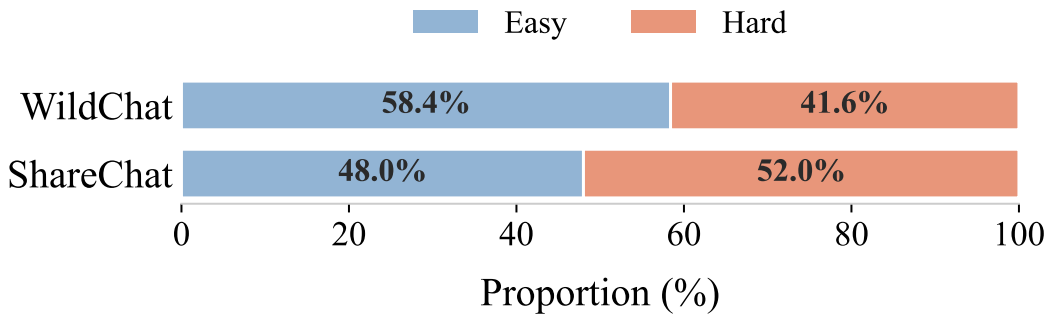


Figure 8: Distribution of difficulty labels across the curated NQP-Wild and NQP-Share datasets.

C Curation Pipeline Attrition

Table 8 reports the number of samples retained at each pipeline stage for the two public subsets.

Table 8: Filtering attrition for the two public NQP-Bench subsets. "Ret.%" is the retention rate relative to the previous stage. Truncation mining recovers predictable prefixes from the screening DROP pool.

Stage	NQP-Wild		NQP-Share	
	#Samples	Ret.%	#Samples	Ret.%
Raw corpus	3,199K	–	130K	–
Stage I (heuristic)	148K	4.6	30.5K	23.6
Stage II (GLM-5)	29.2K	19.7	6.1K	20.0
Stage III (Gemini)	21.0K	72.0	4.7K	77.2
+ Trunc. mining	+6.7K	–	+1.2K	–
Final	18.3K	–	1.9K	–

Stage I removes approximately 95.4% of WildChat examples through the combined heuristics, which include English-only filtering. Stage II forwards approximately 20% of the heuristically filtered pool; the remaining approximately 80% fail its combined safety, predictability, clarity, and repetition criteria. Retention rates summarize each stage as a whole. Stage III retains 72–77% of forwarded samples after review by Gemini 2.5 Pro. After pooling Stage III KEEP and truncation-mined samples, a final quality pass removes near-duplicates (by embedding similarity) and over-length samples, yielding the final benchmark, which is then split into training and test sets stratified by difficulty (Appendix B).

D Case Study

Figure 4 shows a 13-turn NQP-Wild conversation spanning nutrition, health, cooking, and skin collagen. The recorded next query, "Which foods are best to increase collagen production?", links the earlier nutritional interest to the recent collagen topic.

A Current-turn model lacks the earlier food thread. Full-history retains it alongside recipe corrections, numbness symptoms (turns 3–5), and lengthy assistant responses. The recursive memory preserves the trajectory *food/nutrition* $\rightarrow$ *health* $\rightarrow$ *cooking* $\rightarrow$ *skin-health/collagen*, compressing 17.6K characters into 338 characters. Its top candidate matches the recorded intent. The figure abbreviates user queries and presents four representative memory snapshots. In this example, an earlier nutritional interest helps interpret the subsequent collagen-related query. Character compression in this example is distinct from the aggregate token accounting in Section 4.6.

E Additional Baselines

We compare five history interfaces with Gemini-3.1-Pro on NQP-Wild. All use zero-shot prompting, so this comparison examines the interfaces without task-specific training.

- **Sliding-window** (w=3): The model receives only the most recent $w = 3$ user–assistant exchanges as input, discarding all earlier turns.
- **Summarize-then-predict**: At each turn, the model first generates a free-form summary of the full conversation history in a single pass, then predicts the next query conditioned on this summary.

Table 9: Comparison with additional baselines on NQP-Wild (Gemini-3.1-Pro, zero-shot). All methods use the same model without task-specific training, keeping the backbone fixed across history interfaces.

Method	Judge	Human
Current-turn	45.53	45.30
Sliding-window (w=3)	46.14	46.40
Summarize-then-predict	47.42	47.80
Full-history	48.81	49.90
Ours	**50.72**	**51.10**

Judge and Human scores give the same ordering in Table 9: Ours > Full-history > Summarize-then-predict > Sliding-window > Current-turn. Sliding-window (w=3) improves the Judge score over Current-turn by $+0.61$.

Summarize-then-predict improves on Current-turn by $+1.89$ Judge points and falls below Full-history by 1.39 points. Our recursive memory outperforms Full-history by 1.91 and Summarize-then-predict by 3.30 Judge points under the same zero-shot evaluation setting. Section 4.2 extends this comparison to learned-summary controls.

F Hyperparameter Sensitivity

We ablate inference-time hyperparameters (k and N) using Gemini-3.1-Pro in a zero-shot setting on NQP-Wild, isolating the effect of each parameter from task-specific training. We separately ablate the training-time reward weight λ, training a Qwen model independently for each value.

Memory budget k.

Table 10: Effect of memory budget k on NQP-Wild (Gemini-3.1-Pro, zero-shot).

k (tokens)	Judge	Human	Avg. Input Tokens/Turn
100	47.80	48.20	∼250
200	49.35	49.70	∼350
500	**50.72**	**51.10**	∼650
800	50.50	50.90	∼950
1,000	50.25	50.60	∼1,150

Judge scores increase from 47.80 at k=100 to 50.72 at k=500, then decline slightly at the two larger budgets tested. We use k=500 as the default operating point. The token column reports input context at each memory budget: even at k=1,000, the reported input is approximately 1,150 tokens per turn, compared with approximately 14,000 for Full-history at turn 14.

Number of candidate predictions N.

Table 11: Best-of-N scores on NQP-Wild as the candidate count varies (Gemini-3.1-Pro, zero-shot). Bold marks the default setting (N=3).

N	Judge	Human
1	49.15	49.80
2	50.22	50.70
3	**50.72**	**51.10**
5	50.85	51.30

With N=1, the Judge score is 1.57 points below the default N=3 setting, which permits broader candidate-set coverage. The gain from N=2 to N=3 is moderate (+0.50), and N=5 offers only marginal further improvement (+0.13), supporting three candidates as the selected operating point.

Reward weight λ. Since λ is a training-time parameter, each value below corresponds to an independently trained model (RL-trained Qwen on NQP-Wild).

Table 12: Effect of reward weight λ on NQP-Wild (RL-trained Qwen).

λ	Judge	Human
0.5	44.85	45.60
0.7	45.48	46.30
0.9	**46.00**	**47.30**
1.0	45.62	46.60

Among the tested settings, λ=0.9 gives the highest Judge and Human scores. Removing the format component (λ=1.0) lowers the Judge score from 46.00 to 45.62, while λ=0.5 gives 44.85. These observed scores motivate the default choice of λ=0.9, which retains both prediction and format components in the training reward.

G Uncertainty of Aggregate Scores

Table 13 reports approximate 95% intervals for the mean Best-of-3 Judge score on the 2,354-example NQP-Wild test set. The estimand is the mean reference-alignment score of each evaluated model on this test distribution.

Table 13: Approximate marginal 95% intervals for NQP-Wild mean Judge scores.

Method	Mean	Approx. 95% interval
Current-turn	44.02	[42.67,45.37]
Full-history	44.35	[43.00,45.70]
OnePred	46.00	[44.65,47.35]

These marginal intervals describe uncertainty in each model's test-set mean. Inference about a difference between methods requires the paired scores, since all methods are evaluated on the same examples. Accordingly, the table provides neither a paired significance test nor an estimate of variability across training runs. The learned-summary comparisons are reported as point estimates.

H JUDGE VALIDATION

Scoring and ambiguity. Judges score each candidate against the reference on the five-level intent rubric (Table 14). The score is normalized as $S(\hat{q}, q) = 25(s - 1)$ for rating $s \in \{1, \ldots, 5\}$. This rubric is used consistently across the reported experiments. The two highest levels both concern intent agreement, making their boundary sensitive to annotation judgments. A low score can reflect a plausible but unobserved continuation, as well as an incorrect prediction.

Table 14: Intent-rating rubric. Normalized scores are graded agreement with one reference, not accuracy or calibrated probabilities.

Rating	Criterion
5	Intent hit: same core intent, equivalent phrasing
4	Highly aligned: same intent, different phrasing
3	Topic related: shared topic, different specific question
2	Slightly related: same domain, different direction
1	Irrelevant: no meaningful connection

We examine agreement between the scoring procedures on the 250 human-annotated samples per subset. We report inter-annotator agreement and human–judge correlation to characterize consistency on this shared sample.

Human inter-annotator agreement. Five expert annotators independently scored each sample on the same 5-point rubric used by the LLM judges. Fleiss' κ across the five annotators is 0.72 (NQP-Priv), 0.75 (NQP-Wild), and 0.70 (NQP-Share), indicating substantial agreement within these samples.

Human–LLM correlation. We compute Spearman's ρ between the per-sample LLM-judge score (majority vote of three judges) and the corresponding human score (majority vote of five annotators) on the shared 250-sample subsets. The correlation is 0.78 (NQP-Priv), 0.81 (NQP-Wild), and 0.76 (NQP-Share), showing strong sample-level agreement beyond the ranking-level consistency reported in §4.2.

LLM inter-judge agreement. Fleiss' κ among the three LLM judges (Claude Opus 4.7, GPT-5.5, Gemini-3.1-Pro) is 0.68 (NQP-Priv), 0.71 (NQP-Wild), and 0.66 (NQP-Share). The moderate-to-substantial agreement supports the use of majority vote to aggregate their ratings.

Human validation of benchmark curation. We audit curation decisions against human predictability judgments. Three human annotators independently reviewed 400 randomly sampled instances from the curation candidate pool (200 retained KEEP instances and 200 rejected DROP instances) and judged whether the target query was predictable from the conversation context. Human–LLM agreement on the KEEP/DROP decision reached 92% (Cohen's $\kappa = 0.83$) on this balanced audit sample. This analysis measures consistency of curation decisions. The overlapping model families in curation and evaluation, and the reuse of the evaluation ensemble for RL rewards, remain sources of shared bias.

I SCORE DISTRIBUTION ANALYSIS

Table 15 shows the full rubric-level score distribution on NQP-Wild using the recursive memory interface, comparing an untrained base model against the RL-trained model and a frontier API.

Table 15: Score distribution (%, rounded) by rubric level on NQP-Wild. All rows use recursive memory. 1 = Irrelevant, 2 = Slightly related, 3 = Topic related, 4 = Highly aligned, 5 = Intent hit.

Model	1	2	3	4	5	Mean
Base Qwen	26	24	25	16	9	39.57
RL Qwen	21	22	23	20	14	46.00
Gemini-3.1-Pro	15	22	25	22	16	50.72

The RL-trained model assigns less mass to the two lowest rating levels and more to the two highest levels than the base model. Approximately 34% of RL predictions receive a rating of at least 4, compared with 25% for the base model and 38% for Gemini-3.1-Pro. These proportions measure alignment with the recorded continuation; user acceptance and downstream utility are separate outcomes.

J FINE-GRAINED EVALUATION AND ERROR ANALYSIS

Table 16 stratifies performance on the NQP-Wild test set by difficulty and intent transfer paradigm. We describe the observed score differences and propose hypotheses about memory loss and continuation ambiguity for future analysis of individual trajectories.

Fine-grained performance breakdown. OnePred outperforms Full-history on Deepening, Application, and Challenge within both difficulty strata. Table 16 reports each stratum separately. The largest gains occur on Easy $\times$ Deepening ($+5.1$) and Hard $\times$ Challenge ($+5.7$). Associated Shift reverses the ordering: Full-history leads by 3.3 points on Easy examples and 1.8 on Hard examples. The deficit on Associated Shift motivates the detail-loss hypothesis discussed below.

Table 16: Performance (LLM Judge, 0–100 scale) stratified by difficulty and intent transfer paradigm on the 2,354-example NQP-Wild test set (RL-trained Qwen). Bold indicates the better method.

Difficulty	Method	Intent Transfer Paradigm			
		Deep.	Appl.	Assoc.	Chall.
Easy	Full-hist	45.1	42.0	**50.1**	44.3
	Ours	**50.2**	**44.6**	46.8	**45.1**
Hard	Full-hist	44.6	38.8	**44.5**	39.5
	Ours	**45.8**	**40.1**	42.7	**45.2**

Detail loss under compression. The bounded memory is optimized to retain high-level intent trajectories, but fine-grained details (e.g., specific numbers, exact constraints, or minor sub-topics) may be compressed away if the model judges them less salient at the time of the update. Losing a detail needed for the next query could reduce prediction quality. The largest reported deficit occurs on Easy samples with Associated Shift, where Full-history outperforms our method by $+3.3$ (50.1 vs. 46.8 in Table 16), consistent with this hypothesis.

Hypothesized failure patterns. Two hypotheses guide further analysis of individual prediction trajectories: (1) *detail-loss errors*, where the memory captures the overall intent trajectory but drops a specific detail that the next query depends on; and (2) *ambiguity errors*, where the conversation state genuinely supports multiple plausible next queries and the model's top candidates diverge from the ground truth despite being contextually reasonable.

K SUPPLEMENTARY ROBUSTNESS RESULTS

We examine held-out judges, rejected examples, and latency with prefix caching. The rejected subsets provide category-specific comparisons alongside the retained benchmark. The separate deployment-log evaluation is reported in Section 4.4.

Table 17: Held-out-judge scores on NQP-Wild. Model names are the recorded evaluation-platform identifiers. These judges were not used for rewards, but Qwen and GLM families overlap with the backbone and curation models.

Method	Qwen3.7-Max	GLM-5.2	Average
Current-turn	43.71	43.86	43.79
Full-history	44.02	44.15	44.09
OnePred	45.42	45.58	45.50

Table 18: Scores on selected categories of rejected examples. Results describe these subsets rather than the unfiltered source population; subset sizes and population weights are not reported.

Evaluation subset	Current-turn	Full-history	OnePred
Retained predictable	44.02	44.35	46.00
Rejected: random topic jumps	24.86	25.31	25.08
Rejected: unseen private information	23.91	24.07	23.84
Rejected: ambiguous target	29.36	29.91	30.05

Table 19: Qwen3-8B latency in seconds with batch size 1 and greedy decoding, with and without prefix caching. These preliminary measurements describe model-inference latency in the profiled setting.

Turns	Full-history	With cache	OnePred	With cache
2–5	2.41	1.58	0.94	0.90
6–9	4.83	2.91	1.03	0.97
≥ 10	7.84	4.31	1.12	1.04

L MODEL SCALING

Table 20: Scaling on NQP-Wild. All sizes use the recursive-memory interface; the two rightmost columns add two-stage RL.

	Base		RL-trained	
Size	Judge	Human	Judge	Human
1.7B	29.24	30.50	36.54	36.90
4B	34.07	33.90	42.34	42.10
8B	39.57	40.40	46.00	47.30

Across Qwen3 sizes 1.7B, 4B, and 8B, recursive-memory judge scores rise from 29.24/34.07/39.57 before RL to 36.54/42.34/46.00 after RL (Table 20). The Judge score gains from RL are 7.30, 8.27, and 6.43 points for the three model sizes. Human scores also increase with model size and improve after RL. The gains thus occur at all three tested sizes within the Qwen3 family. Their magnitude varies with model size, with the largest absolute gain at 4B.

M Deployment-Log Evaluation Protocol

Evaluation set. The evaluation in Section 4.4 uses 2,000 prediction points from deployment conversations, with a shared subset of 250 examples for human evaluation. Unlike NQP-Bench, this test set is not filtered according to the predictability of the recorded next query. It therefore evaluates performance beyond the benchmark's target-aware selection criterion. The sampling unit is a prediction point: a conversation prefix together with its observed next user query. The recorded continuation serves as the reference for offline scoring.

Model comparison. We evaluate Current-turn, Full-history, Summary-memory RL, and OnePred using the trained Qwen3-8B models. Each method generates three candidate queries under greedy decoding. Current-turn conditions on the latest exchange; Full-history uses the accumulated dialogue; Summary-memory RL and OnePred use their respective recursively updated textual states. The scoring rubric and primary judge ensemble are the same as in Section 4.1.

Metrics. For prediction point i, let S_{ij} denote the normalized intent-alignment score of candidate j against the observed continuation. We report

$$\overline{S}_{\mathrm{Top1}} = \frac{1}{n}\sum_{i=1}^{n} S_{i1}, \qquad \overline{S}_{\mathrm{Best3}} = \frac{1}{n}\sum_{i=1}^{n} \max_{j\in\{1,2,3\}} S_{ij}, \qquad n = 2{,}000. \tag{6}$$

Human Best-of-3 is evaluated on the shared 250-example subset using the same five-level rubric. Top-1 scores the first candidate in the three-candidate output. Best-of-3 measures coverage of the observed intent and requires the future reference to select the highest-scoring candidate.

Interpretation. Table 6 reports offline reference alignment on the sampled deployment conversations. The comparison tests whether the ordering of methods extends beyond predictability-selected examples. The outcomes are offline query-alignment scores. NQP-Bench and the deployment sample differ in conversation source, sample composition, and selection policy, so their absolute scores characterize their respective evaluation settings.